

Avaliativa: Implementações de Pilha

Objetivo: Capacitar ao aluno desenvolver as operações básicas de pilha, utilizando as implementações estática/sequencial e dinâmica/encadeada.

Exercícios:

1. Implementar, utilizando a alocação **estática** e o acesso **sequencial**, o TAD pilha com no máximo 10 *funcionários*, sendo que para cada funcionário deve-se guardar as seguintes informações: matrícula (inteiro longo), nome (com no máximo 30 caracteres), setor ('F' - fábrica, 'A' - administrativo, 'C' - comercial, ou 'G' – gerência) e o salário. Além das operações básicas (*cria_pilha*, *pilha_vazia*, *pilha_cheia*, *push*, *pop* e *get_topo*), o TAD também deve contemplar a operação *apaga_pilha*, que elimina a pilha passada como entrada, caso ela exista; *esvazia_pilha*, que retorna a pilha de entrada para o estado de vazia; e *tamanho_pilha*, que retorna o tamanho da pilha passada como entrada. Por fim, também desenvolva um programa aplicativo que permita ao usuário criar uma pilha, inserir e remover elementos, bem como ver o elemento que está no topo da pilha, apagá-la, esvaziá-la e imprimi-la, desde que ela já tenha sido criada.

OBS: a impressão de uma pilha deve ser feita da base para o topo e utilizar APENAS as operações básicas descritas acima. Apesar de menos eficiente, é o que o desenvolvedor da aplicação pode fazer no cenário descrito neste exercício, dado que NÃO existe a operação *get_elem_pos* (essa operação não respeita a política de acesso LIFO).

2. Implementar, utilizando a alocação **dinâmica** e o acesso **encadeado**, o TAD pilha de *strings*, sendo que cada *string* pode ter no máximo 20 caracteres (não considerando o '\0'). Esse TAD deve ter as mesmas operações da questão anterior, exceto pela *pilha_cheia* e *esvazia_pilha*, que não se aplicam. O programa aplicativo também será o mesmo.